应用于机器人路径规划的双向时效 A*算法 *

高民东，张雅妮，朱凌云†

(重庆理工大学 计算机科学与工程学院，重庆 400054)

摘　要：针对时效 A*算法为了大幅减少算法时间，导致路径规划长度增加和路径锯齿过多的问题，提出一种改进的双向时效 A*算法，该方法将从起点和终点同时运行时效 A*算法寻找路径，并采用多近邻栅格距离计算方案；同时，根据不同环境地图对传统 A*算法、时效 A*算法和双向时效 A*算法运行结果进行对比研究及分析；最后，制定算法时间、路径长度两个指标来评判算法的优劣。实验结果显示，双向时效 A*算法相对于传统 A*算法，算法时间最大减少 76.8%，相对于时效 A*算法，时间最大减少 55.4%，并解决了时效 A*算法规划路径距离增加、路径不够平滑的问题。

关键词：路径规划；启发式算法；时效 A*算法

中图分类号：TP301.6

Bidirectional time-efficient A* algorithm for robot path planning

Gao Mindong, Zhang Yani, Zhu Lingyun†

(*College of Computer Science & Engineering*, *Chongqing University of Technology*, *Chongqing* 400054, *China*)

Abstract: As for the Time-Efficient A* algorithm significantly reduces the algorithm time, lead to the problem of increase path planning length and path sawtooth. This paper proposed an improved Bidirectional Time-Efficient A* Algorithm, which will run the A* algorithm from the start and the end point to find the path and apply the multiple neighbor grid distance computation scheme. At the same time, according to different environment map, this paper compared and analyzed the A* algorithm, Time-Efficient A* algorithm and Bidirectional Time-Efficient A* algorithm. Finally, the index of algorithm time and path length will be chosen as two indicators to evaluate the merits and demerits of the algorithm. Experimental results show that compared with with traditional A* algorithm, the Bidirectional Time-Efficient A* decreases the time of the algorithm by 76.8% at the maximum, and reduces the maximum time by 55.4% compared with Time-efficient A* algorithm. It also solves the problem that the path distance of the Time-Efficient A* algorithm is augmented and the path is not smooth enough.

Key Words: path planning; heuristic algorithm; time-efficient A* algorithm

0　引言

路径规划技术是机器人研究的一个重要领域，是实现轮式机器人自主定位与导航的关键技术之一。路径规划主要任务是使机器人能够安全避开各种环境中的障碍物，并以最短且最优的路径到达所指定的终点。路径规划技术根据环境的已知程度可以分为基于模型的全局路径规划和基于传感器的局部路径规划，根据机器人周围环境是否变化可分为静态路径规划和动态路径规划[1]；在动态路径规划研究中，人工势场法具有同时处理静态、动态环境的优点，但在路径规划过程中容易陷入局部最小值，目前已有很多相关领域专家针对此问题展开了广泛的研究，如将机器人相对加速度引入引力势场函数的思想[2]；智能仿生路径规划具有搜索快、参数少，易于实现的优点，郑健，黄敏等人曾通过引入遗传交叉因子、精英保留策略和动态侦查蜂机制对传统人工蜂群算法进行改进，并将其应用于指路标志系统中[3]。

在全局静态路径规划算法中，因为 A*算法的成熟性及其稳定性，诸多学者对 A*算法进行了多种改进研究：Sariff 等人曾对不同的可视图法与 A*算法结合后的算法进行对比研究，并进行综述性分析[4]；针对地图构建方面的改进， Wang 等人对路径规划在城市地图中进行应用研究，提出基于 A*算法的分层 A*算法，该算法通过对城市地图进行分层处理，然后对不同层次地图进行 A*算法路径规划，有效的较少了算法的运行时间[5]；Lv 等人提出以拓扑模型方式建立地图，并以 A*算法来解决自主地面车辆的路径规划问题。该方法解决了多个自主车辆路径规划的问题，并且稳定性较高[6]；李冲等人提出了一种新的单边矩形扩展 A*算法，该算法采用受迫扩展规则，用单条公共边取代相邻矩形的两条重复边来提高算法效率[7]。刘大瑞等人基

基金项目：国家自然科学基金资助项目（61502064）；重庆市教委科学技术研究项目（KJ1709206）；重庆理工大学研究生创新基金资助项目（YCX2016228）

作者简介：高民东（1994-），男，硕士研究生，主要研究方向为机器视觉、机器人路径规划；张雅妮（1994-），女，硕士研究生，主要研究方向为机器视觉、图像处理；朱凌云（1969-），男（通信作者），副教授，博士，主要研究方向为计算机视觉智能感知技术（zlycqu@163.com）．

于代价向量多目标 A*算法进行改进，提出 MOVA*算法，并将该算法成功应用于 NPC 路径规划中[8]。

本文主要针对全局静态路径规划算法进行改进研究。Guruji 提出基于 A*算法改进的时效 A*算法，该算法通过采用斜率检测计算方式来减少传统 A*算法的近邻搜索次数，从而达到提高算法效率的目的，但却增加了规划路径的长度，且路径不够平滑[9]；本文针对该算法继续进行改进，首先，通过在起点、终点同时运行时效 A*算法，使得路径搜索更快，算法效率进一步提高；其次，在解决路径不够平滑问题时，从辛煜，梁华为等人提出的无限邻域的 A*算法思想[10]以及 Fernandes 等人提出的修改传统 A*算法的近邻数来使路径规划适用于多边形障碍物情况的思想[11]中研究新的近邻搜索矩阵改进思路，采用改进的多近邻搜索方案来替换原 8 近邻搜索方案[12]；最后，本文针对对不同环境地图对三种路径规划算法进行仿真实验结果对比与分析，结果显示，该方案极大的改善了时效 A*算法因减少算法运行时间而导致路径长度增加以及路径不够平滑的问题。

1　时效 A*算法

传统 A*算法采用启发式函数路径规划思路，其主要思想是：从起点开始，检测当前栅格周围八个近邻栅格的 $f(n)$值，并选取周围八个近邻栅格 $f(n)$值最小的栅格为下一个，继续进行检测，直到检测范围到达终点为止。其中，$f(n)$值的计算公式如下：

$$f(n)=g(n)+h(n) \tag{1}$$

其中：$g(n)$ 是当前栅格到第 n 个 n 近邻栅格的距离耗费，$h(n)$ 是第 n 个近邻栅格到终点的距离耗费。栅格之间的距离计算方法有很多种，如欧氏距离、马氏距离和曼哈顿距离等。本文传统 A*算法采用欧式距离计算 $g(n)$ 和 $h(n)$ 的值，欧式距离计算公式如下所示：

$$g(n) \text{ or } h(n)=\sqrt{(x_n-x_o)^2+(y_n-y_o)^2} \tag{2}$$

时效 A*算法思想是在传统 A*算法中加入了斜率检测阶段，从而减少了传统 A*算法的运行时间。与传统 A*算法不同，时效 A*算法主要在临近碰撞物前才会去计算当前栅格周围八个近邻栅格的启发式函数 $f(n)$ 值，而不需要从一开始就执行 A*算法的近邻栅格计算步骤。时效 A*算法主要分为五个阶段：

a)输入阶段，主要输入机器人的运行环境地图和机器人的起点、终点。运行环境地图主要为经过栅格法处理过后的二进制位图，其黑色区域代表障碍物，白色区域代表可通行区域。

b)接近检测阶段，该阶段会检测起点和终点的距离，如果距离较短，则意味这接近条件满足，然后执行 A*算法，否则继续下一阶段。

c)斜率检测阶段，该阶段将会以起点和下一目标点为终点计算斜率，并与起点至终点计算的初始斜率进行比较，如果斜率统一，则投射到该目标点，如果斜率不统一，则根据斜率是否大于或小于统一值来投射下一目标点到任意角度。

d)转换阶段，当目标点 S 临近障碍物后，标记目标点为 *SwitchS*，同时将 *SwitchS* 作为新的起点，并从该点进行 A*算法继续规划路径。

e)结合阶段，当 A*算法将终点加入关闭列表后，即可得出一条规划路径，结合斜率检测得出的路径结果以及 A*算法得出的路径结果即可获得最终路径规划结果。

2　双向时效 A*算法

双向时效 A*算法流程与时效 A*算法大致相同，其不同之处在于对时效 A*算法斜率检测阶段和转换阶段的改进；本章主要讨论本文改进算法的算法流程以及本文提出的多近邻栅格改进方案。

2.1　算法近邻栅格改进

首先针对时效 A*算法中的近邻搜索矩阵进行改进，即转换阶段中的执行的 A*算法固定 8 个近邻栅格搜索方式改进为 16、32 和 56 近邻栅格搜索方式。定义 M_n 代表 n 个近邻搜索矩阵，同时，矩阵值中的元素 $M_n(x,y)$ 代表第 x 行，第 y 列的近邻栅格，矩阵中心 $M_n(x+1/2,y+1/2)$ 为开启列表中的目标点，根据以上定义，可以得出原始算法的 8 近邻栅格搜索矩阵为

$$\boldsymbol{M_8}=\begin{bmatrix}1&1&1\\1&0&1\\1&1&1\end{bmatrix} \tag{3}$$

其中:当 $M_8(x,y)=1$ 代表该栅格为需要进行距离计算的近邻栅格且存在与开始列表中，当 $M_8(x,y)=0$ 代表该栅格不需要进行距离计算，且在关闭列表中。

同理可得 16 近邻栅格搜索矩阵如式(4)所示。

$$\boldsymbol{M_{16}}=\begin{bmatrix}0&1&0&1&0\\1&1&1&1&1\\0&1&0&1&0\\1&1&1&1&1\\0&1&0&1&0\end{bmatrix} \tag{4}$$

对比 M_8 和 M_{16} 可以看出，采用 M_8 作为搜索矩阵的路径规划算法具有 8 个备选方向，而采用 M_{16} 作为搜索矩阵的路径规划算法具有 16 个备选方向，使得路径规划方向选择更加多样化。同理可得 32 近邻的搜索矩阵 M_{32} 和 56 近邻的搜索矩阵 M_{56}。

根据以上公式可以看出，多近邻搜索矩阵 M_{16}、M_{32} 和 M_{56} 比传统算法近邻搜索矩阵 M_8 拥有更多的备选方向，同时在近邻搜索过程中，更更大范围的进行启发式函数的遍历计算，从理论上不仅可以使得路径规划过程更加迅速，还能减少路径规划路线结果的锯齿，使其更加平滑。

2.2　算法改进流程

同时效 A*算法一样，双向时效 A*算法同样包含五个阶段：输入阶段、接近检测阶段、斜率检测阶段、转换阶段、结合阶段。但在斜率检测阶段，双向时效 A*算法是从起点和终点同时进行斜率检测，从而在原算法基础上再次减少了算法的遍历次数以及算法运行时间。其算法流程如图 1 所示。

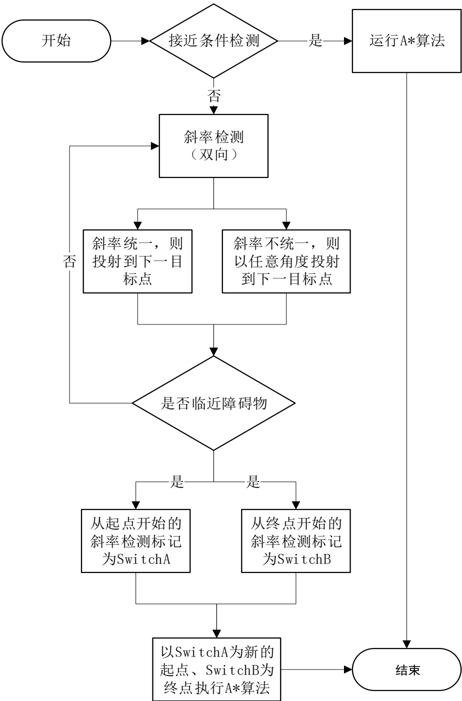


图 1　双向时效 A*算法流程图

从图 1 中可以看出，在斜率检测阶段，算法首先计算起点与终点的初始斜率，然后同时计算并比较起点与下一目标点斜率，终点与下一目标点与斜率与初始斜率的大小，如果斜率统一，则分别投射至下一目标点，如果不统一，则根据斜率是否大于或小于统一值来投射下一目标点到任意角度。

在转换阶段，会存在两个目标点临近障碍物，一个是从起点投射的目标点 A，另一个是从终点投射的目标点 B，然后标记目标点 A 为新的起点 *SwitchA*，目标点 B 为新的终点 *SwitchB*，并以 *SwitchA* 和 *SwitchB* 为输入变量转而执行 A*路径规划算法。

3　仿真结果与分析

本节主要对传统 A*算法、时效 A*算法、双向时效 A*算法在不同地图环境下进行对比实验，实验平台为 MATLABr2016a，以路径规划结果、路径长度、算法运行时间为评价指标进行算法优劣的评价。本文首先实现了三种算法，并运用实现的算法在不同地图环境下的路径规划结果进行实验得出数据；然后对三种算法的路径长度、算法运行时间等实验结果进行对比分析；最后根据不同近邻下双向时效算法的实验结果，并结合其余两种算法进行研究与分析。

3.1　路径规划结果对比分析

首先对三种算法在不同地图环境下的路径规划结果进行实验与分析，本文设置五种地图环境，其中前三种为小地图环境，包括单障碍物、多障碍物和圆形障碍物等情况，其大小为 100×100 dpi；后两种为大地图环境，包括圆形障碍物和混合障碍物情况，其大小为 500×500 dpi。设置机器人起点为[10, 10]，小地图环境终点为[95, 95]，大地图环境终点为[450, 450]。其路径规划实验结果如图 2 所示。

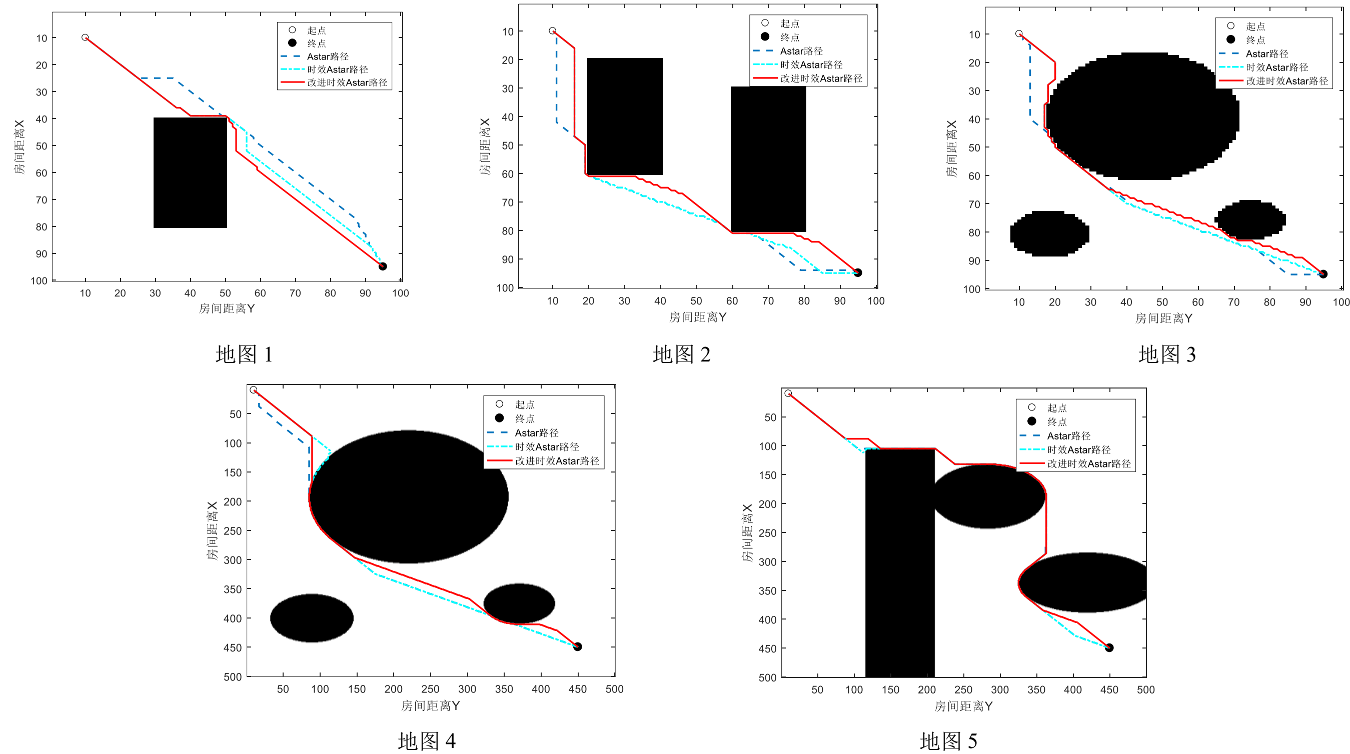


图 2　不同算法路径规划结果图

图 2 中显示了传统 A*算法、时效 A*算法、双向时效 A*算法的路径规划对比结果，在小地图环境下，图 2(a)(c)的改进算法规划路径结果比其余两种算法规划路径结果更优越，并减少了不必要的行走路径；图 2(b)中改进算法规划路径与其余两种算法规划路径略优，且锯齿过多。在大地图环境下，图 2(d)的改进算法规划路径比其余两种算法更优越，且弯折较少，虽然

时效 A*算法规划路径亦合理，但规划路径距离障碍物过近，不适合机器人行走；图 2(e)的改进算法路径并避开了方形障碍物的转折点，使得比其余两种算法规划路径更加合理。

图 2 中改进算法路径在大多数地图环境下比其余两种算法更加合理、有效，但是其锯齿仍然过多，故通过第二步改进，即替换原搜索矩阵 M_8 为近邻搜索矩阵 M_{16}、M_{32} 和 M_{56}，其实验结果如图 3 和 4 所示。

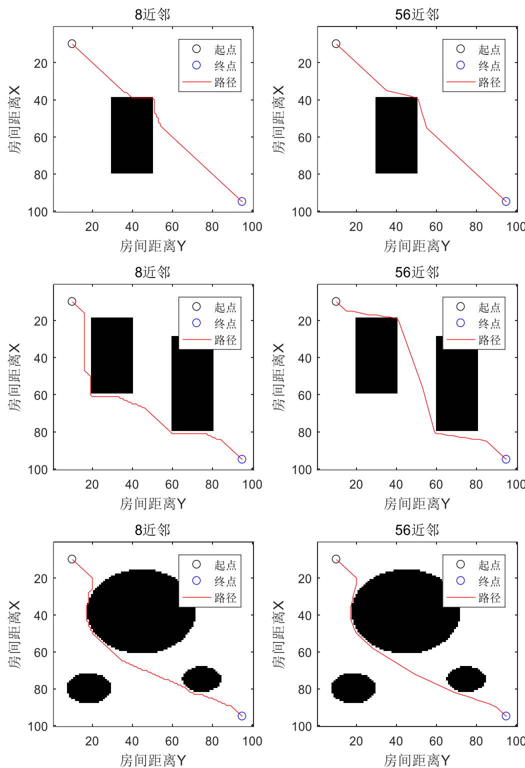


图 3　多近邻路径规划结果对比（小地图环境）

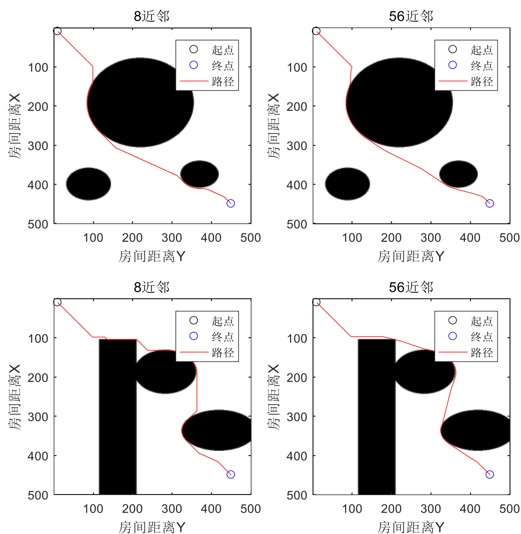


图 4　多近邻路径规划结果对比（大地图环境）

因篇幅有限，图 3 和 4 仅对比双向时效 A*算法（8 近邻）和双向时效 A*算法（56 近邻）的路径规划结果。从图 3 和 4 可以看出，经过替换多近邻搜索矩阵后的双向时效 A*算法比原双向时效 A*算法路径更加平滑，无过多弯折，并有效的避开了障碍物，在图 3 的地图二环境中，多近邻改进后的算法重新规划处一条与原算法完全不同的更优路径，亦达到了路径平滑的要求。综上所述，本文改进算法可以完整的规划出一条比其余两种算法更优更平滑的路径结果。

3.2　算法效率与路径长度对比分析

本文以算法效率、路径长度为评价指标来判定本文改进算法的路径规划结果是否达到最优最短路径的要求。通过上节对三种算法的对比实验，可以得出传统 A*算法、时效 A*算法、双向时效 A*算法的算法效率以及路径规划长度结果，如图 5~7 所示。

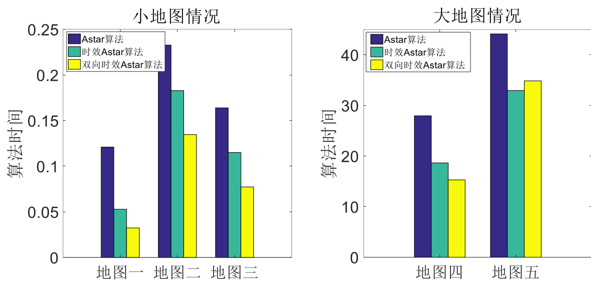


图 5　算法效率对比结果

图 5 显示了三个不同算法的运行时间对比，可以明显看出，传统 A*算法规划路径时间过长，而时效 A*算法在原算法基础上明显减少了算法运行时间，本文改进算法在大多数地图情况下提升了算法效率，但在地图五情况下，算法运行时间较时效 A*算法高。

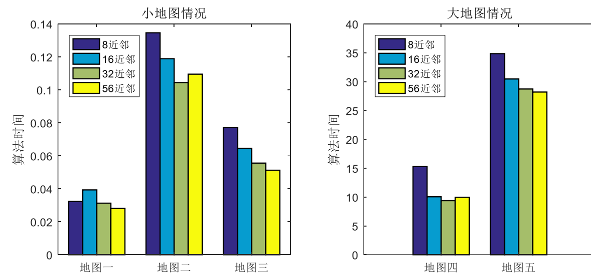


图 6　多近邻算法效率对比结果

通过多近邻搜索矩阵可以使得每一次遍历栅格数增加而减少算法循环次数，从而减少算法运行时间，同时多近邻搜索矩阵提供更多的路径备选方向，使得路径更加平滑，针对改进后的双向时效 A*算法进行替换多近邻搜索矩阵实验，其算法效率实验结果对比如图 6 所示。

从图 6 中可以看出，替换多近邻搜索矩阵后的双向时效 A*算法比前面几种算法效率更高；在某些地图环境下，32 近邻的双向时效 A*算法要比 56 近邻的双向时效 A*算法的算法效率略高，但算法运行时间总体呈下降趋势；在地图五中，多近邻改进后的双向时效 A*算法效率得到了明显提高，比时效 A*算法运行时间更少。

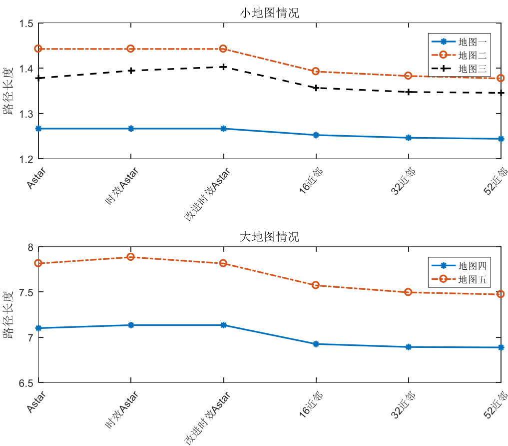


图 7　算法路径长度对比结果

从图 7 可以看出，时效 A*算法在减少传统 A*算法运行时间的过程中，也会导致其路径长度的增加，本文改进算法虽然在某些地图情况下，也会在时效 A*算法规划路径长度结果上进一步增加，但进过替换多近邻搜索矩阵后的双向时效 A*算法，其路径规划长度结果呈下降并逐步稳定趋势；最终结果表明，本文改进算法不仅在算法效率上比时效 A*算法更高，而且还解决了时效 A*算法为了提高算法效率而增加路径规划长度的问题，并小幅减少传统 A*算法的路径规划长度。

3.3　小结

本节对传统 A*算法、时效 A*算法和双向时效 A*算法在五种不同地图环境下，以路径规划结果、算法效率及路径长度为评价指标进行了仿真实验与结果分析，最终得出表 1 的实验结果。表 1 中列举了三种不同算法的算法效率以及路径长度结果对比，并计算了其算法效率减少百分比以及路径规划结果减少百分比。

从表 1 中可以看出，本文改进算法比原作者算法在算法效率上最少提升了 14.4%，最大提升了 55.4%；在规划路径结果中，本文改进算法相对于原作者算法最大减少了 5.2%的路径长度。综上所述，多近邻改进后的双向时效 A*算法在规划路径过程中，相对于其余两种算法，在保证路径最优的情况下，不仅有效、合理的减少了算法运行时间，而且解决了时效 A*算法的路径长度增加问题，在替换多近邻矩阵后的算法路径规划结果有效减少了路径的弯折过多的问题，路径平滑可靠，使得改进后的双向时效 A*算法在路径规划中的应用更加具有可靠性和稳定性。

表 1　算法路径长度对比结果

地图编号	传统 Astar 算法		时效 Astar 算法		双向时效 Astar 算法		结果对比	
	算法效率/s	路径长度/m	算法效率/s	路径长度/m	算法效率/s	路径长度/m	减少时间	减少长度
1	0.121	1.267	0.053	1.267	0.028	1.244	46.8%	1.8%
2	0.233	1.442	0.183	1.442	0.110	1.377	40.1%	4.5%
3	0.164	1.378	0.115	1.394	0.010	1.345	55.4%	3.5%
4	27.949	7.101	18.636	7.134	9.950	6.888	46.6%	3.5%
5	44.130	7.814	32.952	7.884	28.195	7.473	14.4%	5.2%

4　结束语

本文针对时效 A*算法进行改进研究，时效 A*算法在传统 A*算法的基础上使得算法的效率提高，但是也导致了路径距离增大的问题，同时规划路径的弯折过多不够平滑；双向时效 A*算法在原时效 A*算法进行了两处改进：首先改进了算法的流程，使得时效 A*算法从起点、终点开始同时搜寻路径，从而进一步提高了算法效率；然后替换原 8 近邻搜索矩阵为多近邻搜索矩阵，使得路径备选方向增加，路径规划结果弯折明显减少，亦解决了时效 A*算法在路径规划中距离增加的问题。故基于时效 A*算法改进的双向时效 A*算法在路径规划结果、算法效率和路径长度上相较前两种算法都具有明显的优点，其路径规划结果可靠性高，可以合理地应用于机器人路径规划中。

本文下一步将针对以下方向继续进行研究，其一是在距离障碍物多少距离前标记目标点为 *SwitchA* 和 *SwitchB*，此处应可以定义一种评判准则来进行判定；其二是可以考虑不同地图环境来决定不同近邻的搜索矩阵。

参考文献：

[1] 徐飞. 基于改进人工势场法的机器人避障及路径规划研究 [J]. 计算机科学, 2016, 43 (12): 293-296.

[2] 徐腾飞, 罗琦, 王海. 基于向量场的移动机器人动态路径规划 [J]. 计算机科学, 2015, 42 (5): 237-244.

[3] 郑健, 黄敏, 张腾, 等. 求解指路标志指引路径规划问题的改进人工蜂群算法 [J]. 计算机应用研究, 2017, 34 (8): 2355-2359.

[4] Sariff N, Buniyamin N. An overview of autonomous mobile robot path planning algorithms [C]// Proc of the 4th Student Conference on Research and Developmen. 2006: 183-188.

[5] Wang H, Zhou J, Zheng G, et al. HAS: hierarchical A-star algorithm for big map navigation in special areas [C]// Proc of the 5th International Conference on Digital Home. 2014: 222-225.

[6] Lv M, Gao T, Zhang N. Research of AGV scheduling and path planning of automatic transport system [J]. International Journal of Control and Automation, 2016, 9 (4): 1-10.

[7] 李冲, 张安, 毕文豪. 单边矩形扩展 A*算法 [J]. 机器人, 2016, 39 (1):

46-56.

[8] 刘大瑞，钱程，林涛. 基于多目标A*算法的游戏NPC路径规划 [J]. 计算机应用研究, 2014, 31 (8): 2279-2282.

[9] Guruji A K, Agarwal H, Parsediya D K. Time-efficient A* algorithm for robot path planning [J]. Procedia Technology, 2016, 23 (3): 144-149.

[10] 辛煜，梁华为，杜明博，等. 一种可搜索无限个邻域的改进 A*算法 [J]. 机器人, 2014, 36 (5): 627-633.

[11] Fernandes E, Costa P, Lima J, et al. Towards an orientation enhanced A* algorithm for robotic navigation [C]// Proc of IEEE International Conference on Industrial Technology. 2015: 3320-3325.

[12] Ueland E S, Skjetne R, Dahl A R. Marine autonomous exploration using a lidar and SLAM [C]// Proc of the 36th International Conference on Ocean, Offshore and Arctic Engineering-Volume 6: Ocean Space Utilization. American Society of Mechanical Engineers. 2017.